PanLex 3.0: The Database Design

Jonathan Pool
PanLex Project
http://panlex.org
The Long Now Foundation

March 02013

Contents

1. Introduction

A. PanLex is a database that represents assertions about the meanings of expressions. To be specific:

B. **Database**: A relational database (PostgreSQL).

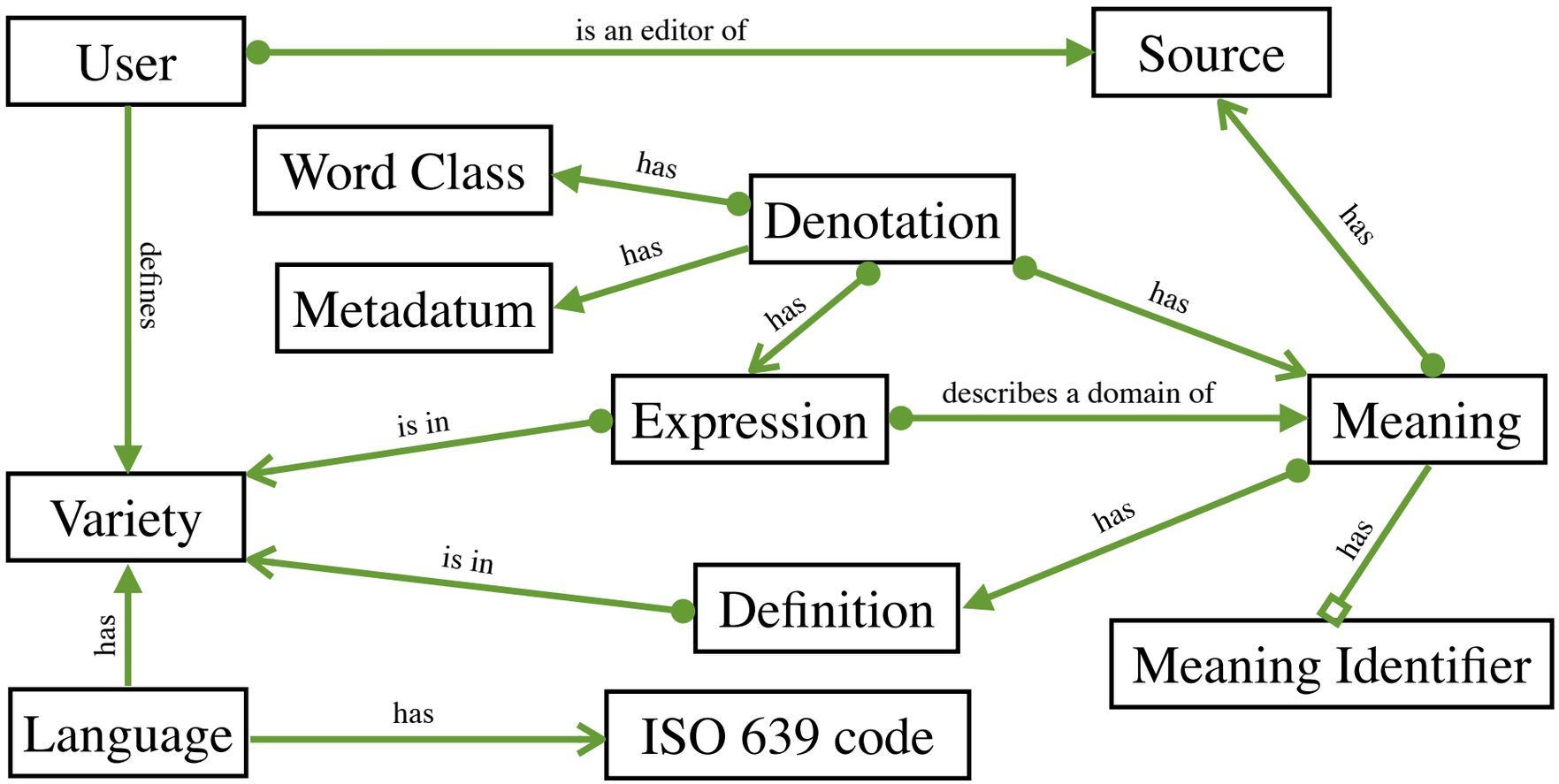
C. **Assertions**: Factual claims, not infallible truths. “A says B = C”, not “B = C”. Assertions are attributed to their makers and can disagree.

D. **Expressions**: Expressions that are lexemes. A lexeme is an entry (word or phrase) in the lexicon of a language. It is represented in a “citation” or “dictionary” form, i.e. as a “lemma”. E.g., “go” is a lemma, “went” is not. A noncompositional phrase (one not interpretable from its parts) is a lexeme: “green thumb” is a lexeme, “green paint” is not.

E. **Meanings**: Expressions have meanings. Expressions that share a meaning are translations or synonyms.

1. Introduction

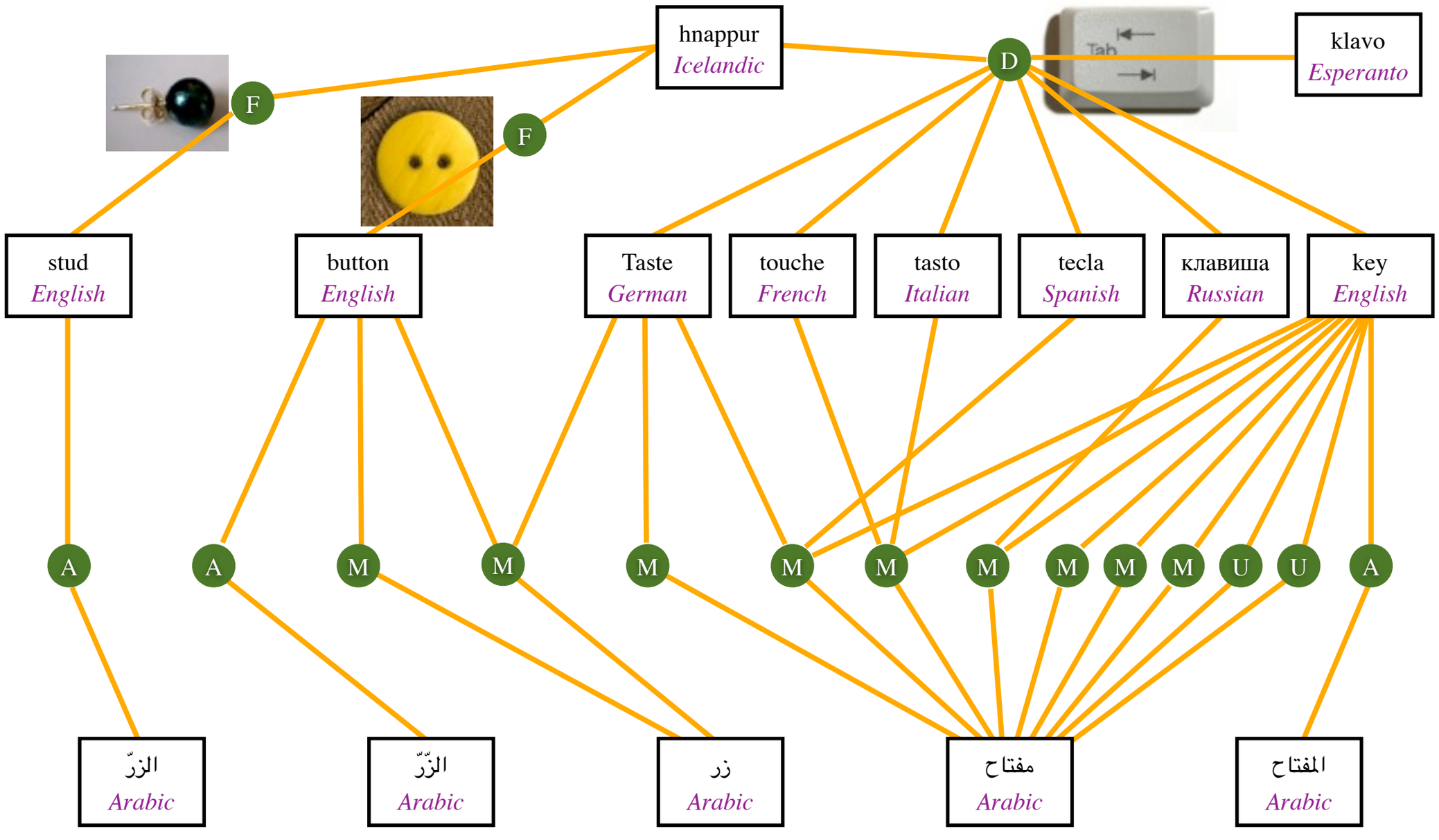
Entities and Relationships in PanLex:
Informal Summary



1. Introduction

Design Motivation

A graph with expression and meaning nodes can represent asserted translations. Searches over the graph can yield inferred translations.



2. Constraints

Sources of assertions vary in detail. PanLex is designed to accept even the simplest assertions, plus additional assertion types from richer sources.

Simple

Basque

Russian

Rich

fouyapin (également "foubap" selon le Dictionnaire de Poullet et al., 1990), aussi "friyapen", avec la variante "penbwa" d'après R. Confiant (dictionnaire en ligne) : arbre à pain, fruit à pain.
Il est intéressant de signaler que si, à l'heure actuelle, ne sont répertoriées aux Antilles dans les dictionnaires courants que ces formes calquées sur le mot français, il existe à la Dominique (île voisine indépendante après avoir été colonie britannique au XIXe et XXe siècle) pour désigner la même réalité dans le créole local, le terme de "yanm-pen" (lit. igname-pain) qui tend à faire de "yanm" un terme générique pour "nourriture". Marcel Fontaine dans son dictionnaire cite également l'usage de "pen-pen" - sans indiquer toutefois s'il faut en rapporter l'usage à un groupe particulier.
De fait "penbwa" (pour arbre à pain) est attesté à Sainte-Lucie (île également indépendante après la colonisation britannique).
Ces mots composés créoles sont particulièrement significatifs et intéressants : ne peut-on pas penser que le créole de la Dominique et le créole de Ste-Lucie nous livrent là un usage "non-contaminé" par le français ? Resterait à chercher si ces formes ont été également attestées en créole au XIXe siècle en Guadeloupe et Martinique.
En Haïti semble attestée la forme (un peu étonnante : on s'interroge sur son origine) de "lam"/"lanm" ou "lam véritab" (cf. in Dictionnaire d'Albert Valdman et al., 1996, mais aussi in Wally R. Turnbull, 2003 : *Creole Made Easy*)

3. Users

Data in PanLex come from **users**.

Table “us”

```
 Column |  Type   |                Modifiers                | Storage  |              Description
--------+---------+-----------------------------------------+----------+---------------------------------
 us     | integer | not null                                | plain    | ID
 dt     | date    | not null default ('now'::text)::date | plain    | enrollment date
 nm     | text    |                                         | extended | name
 al     | text    | not null                                | extended | alias (username)
 sm     | text    |                                         | extended | SMTP (Internet mail) address
 ht     | text    |                                         | extended | HTTP (World Wide Web) address (URL)
 ok     | boolean | not null default false                  | plain    | whether approved
 ad     | boolean | not null default false                  | plain    | whether a PanLex superuser
Indexes:
    "us_pkey" PRIMARY KEY, btree (us) CLUSTER
    "us_al_key" UNIQUE, btree (al)
Referenced by:
    TABLE "au" CONSTRAINT "au_us_fkey" FOREIGN KEY (us) REFERENCES us(us)
    TABLE "auto" CONSTRAINT "auto_tsus_fkey" FOREIGN KEY (tsus) REFERENCES us(us)
    TABLE "lu" CONSTRAINT "lu_us_fkey" FOREIGN KEY (us) REFERENCES us(us)
    TABLE "pw" CONSTRAINT "pw_us_fkey" FOREIGN KEY (us) REFERENCES us(us)
```

4. Sources

Users base assertions on particular source documents, such as dictionaries or thesauri (or their personal knowledge). The combination of user + source document, which has approved any assertion, is the assertion's **source**.

Table "ap"

Column	Type	Modifiers	Storage	Description
ap	integer	not null	plain	ID
dt	date	not null default ('now'::text)::date	plain	registration date
tt	text	not null	extended	label
ur	text		extended	URI
bn	text		extended	ISBN
au	text		extended	author
ti	text		extended	title
pb	text		extended	monograph publisher or serial title, volume, and page range
yr	smallint		plain	year of publication
uq	smallint	not null	plain	quality in editor's judgment (0 to 9)
ui	smallint		plain	numeric ID specified by the user
ul	text		extended	miscellaneous information
li	character(2)		extended	type of offered license
ip	text		extended	summary of intellectual-property claim
co	text		extended	name of apparent intellectual-property claimant
ad	text		extended	SMTP address for licensing correspondence

```
Indexes:
    "ap_pkey" PRIMARY KEY, btree (ap) CLUSTER
    "ap_tt_key" UNIQUE, btree (tt)
Foreign-key constraints:
    "ap_li_fkey" FOREIGN KEY (li) REFERENCES apli(li)
Referenced by:
    TABLE "af" CONSTRAINT "af_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "aped" CONSTRAINT "aped_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)
    TABLE "au" CONSTRAINT "au_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)
    TABLE "av" CONSTRAINT "av_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "mn" CONSTRAINT "mn_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)
```

4. Sources

The license under which a source's documents are published is drawn from a list.

Table "apli"

```
 id | li | pl
 Column |     Type     | Modifiers | Storage  |             Description
--------+--------------+-----------+----------+-------------------------------------
 id     | integer      | not null  | plain    | ID
 li     | character(2) | not null  | extended | 2-character type code
 pl     | text         | not null  | extended | text of the type's PanLex expression
Indexes:
    "apli_pkey" PRIMARY KEY, btree (id) CLUSTER
    "apli_li_key" UNIQUE, btree (li)
    "apli_pl_key" UNIQUE, btree (pl)
Referenced by:
    TABLE "ap" CONSTRAINT "ap_li_fkey" FOREIGN KEY (li) REFERENCES apli(li)
```

```
 id | li | pl
----+----+-----
 10 | nr | pbl  public domain
 20 | cc | ccl  Creative Commons
 30 | rp | ask  request
 40 | gp | gpl  GNU General Public License
 50 | gl | gll  GNU Lesser General Public License
 60 | gd | gfd  GNU Free Documentation License
 65 | mi | mit  MIT License
 70 | co | cpr  copyright
 80 | zz | ali  other
 90 | na | zxx  unknown
```

5. Language Varieties

Each expression is in a language **variety**. It may be the standard variety, a dialect, a script-based variety, a controlled technical variety, etc. Its language is specified with a 3-letter ISO 639 code.

Table “lv”

```
 Column |     Type     |       Modifiers       | Storage  |                Description
--------+--------------+-----------------------+----------+-------------------------------------------
 lv     | integer      | not null              | plain    | ID
 lc     | character(3) | not null              | extended | ISO 639 code
 vc     | smallint     | not null              | plain    | language-specific ID
 sy     | boolean      | not null default true | plain    | whether the variety permits synonymy
 am     | boolean      | not null default true | plain    | whether the variety permits ambiguity
 ex     | integer      | not null              | plain    | expression whose text is the variety’s default name
Indexes:
    "lv_pkey" PRIMARY KEY, btree (lv) CLUSTER
    "lv_lc_key" UNIQUE, btree (lc, vc)
Foreign-key constraints:
    "lv_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex) DEFERRABLE INITIALLY DEFERRED
    "lv_lc_fkey" FOREIGN KEY (lc) REFERENCES lc(lc)
Referenced by:
    TABLE "av" CONSTRAINT "av_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "cp" CONSTRAINT "cp_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    TABLE "cu" CONSTRAINT "cu_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    TABLE "df" CONSTRAINT "df_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    TABLE "ex" CONSTRAINT "ex_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    TABLE "dev.le" CONSTRAINT "le_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    TABLE "lu" CONSTRAINT "lu_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    TABLE "pl1" CONSTRAINT "pl1_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv) ON UPDATE CASCADE ON DELETE CASCADE
```

5. Language Varieties

Example

PanLex has 10 varieties to which ISO 639 code “cmn” (Mandarin) has been assigned.

```
  lv  | lc  | vc | sy | am | ap |     extt
------+-----+----+----+----+----+--------------
 1627 | cmn |  0 | t  | t  |  6 | 简体字
 1628 | cmn |  1 | t  | t  |  6 | 繁體中文
  128 | cmn |  2 | t  | t  |  6 | 官話
 1835 | cmn |  3 | t  | t  |  6 | pīnyīn
 2166 | cmn |  4 | t  | t  |  6 | Muping
 2561 | cmn |  5 | t  | t  |  6 | Xi’an
 3252 | cmn |  6 | t  | t  |  6 | Chengdu
 3253 | cmn |  7 | t  | t  |  6 | Yangzhou
 3254 | cmn |  8 | t  | t  |  6 | Nanjing
 3255 | cmn |  9 | t  | t  |  6 | Ürümqi
(10 rows)
```

Comment: An lc-vc pair (e.g., “cmn, 4”) uniquely identifies a language variety, equivalently to an lv (e.g., 2166).

5. Language Varieties

The language code of each language variety is drawn from a list. It includes all ISO 639-3 individual and macrolanguage codes, all ISO 639-2 collective codes, and all ISO 639-5 codes.

Table “lc”

```
 Column |     Type     | Modifiers | Storage  |                                              Description
--------+--------------+-----------+----------+------------------------------------------------------------------------------------------
 lc     | character(3) | not null  | extended | code
 tp     | character(1) | not null  | extended | code type: “i” = 639-3 ind, “m” = 639-3 macro, “c” = 639-2 coll, “f” = 639-5, “o” = other
Indexes:
    "lc_pkey" PRIMARY KEY, btree (lc) CLUSTER
Referenced by:
    TABLE "i1" CONSTRAINT "i1_iso3_fkey" FOREIGN KEY (iso3) REFERENCES lc(lc)
    TABLE "lv" CONSTRAINT "lv_lc_fkey" FOREIGN KEY (lc) REFERENCES lc(lc)
```

6. Expressions

An **expression** is distinguished by its variety and a textual representation of its lemma.

Table “ex”

```
 Column |  Type   | Modifiers | Storage  | Description
--------+---------+-----------+----------+-------------
 ex     | integer | not null  | plain    | ID
 lv     | integer | not null  | plain    | variety
 tt     | text    | not null  | extended | text
 td     | text    | not null  | extended | degraded text
Indexes:
    "ex_pkey" PRIMARY KEY, btree (ex)
    "ex_lv_key" UNIQUE, btree (lv, tt)
    "ex_lv_idx" btree (lv)
    "ex_td_idx" btree (td)
    "ex_tt_idx" btree (tt) CLUSTER
Foreign-key constraints:
    "ex_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
Referenced by:
    TABLE "dm" CONSTRAINT "dm_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
    TABLE "dn" CONSTRAINT "dn_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
    TABLE "lv" CONSTRAINT "lv_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex) DEFERRABLE INITIALLY DEFERRED
Triggers:
    ex_td BEFORE INSERT OR UPDATE ON ex FOR EACH ROW EXECUTE PROCEDURE tdau()
```

6. Expressions

Comment: Expressions with the same text (lemma) may exist in multiple language varieties. For example, “mata” is an expression in 218 different language varieties in PanLex.

Comment: No two or more expressions in the same language variety may have the same text. For example, “bear” in English is only one expression in PanLex.

Comment: The “td” (degraded text) value, usable in fuzzy comparison and search, is a degradation of the “tt” (text) value. The “td” value applies NFKD normalization to the “tt” value, omits all characters except letters and decimal numbers (those with Unicode General Category Properties Ll, Lo, and Nd), converts all upper-case letters to lower-case, and converts “ı” to “i”. For example, expressions with “tt” values “co-operative”, “coöperative”, “cooperative”, and “Cooperative” all have the “td” value “cooperative”.

7. Meanings

Meanings are source-specific.

Table “mn”

```
 Column |  Type   | Modifiers | Storage | Description
--------+---------+-----------+---------+-------------
 mn     | integer | not null  | plain   | ID
 ap     | integer | not null  | plain   | source
Indexes:
    "mn_pkey" PRIMARY KEY, btree (mn)
    "mn_ap_idx" btree (ap) CLUSTER
Foreign-key constraints:
    "mn_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)
Referenced by:
    TABLE "df" CONSTRAINT "df_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
    TABLE "dm" CONSTRAINT "dm_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
    TABLE "dn" CONSTRAINT "dn_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
    TABLE "mi" CONSTRAINT "mi_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
```

8. Denotations

Denotations are expression-meaning pairs, i.e. assertions that particular expressions have particular meanings.

Table “dn”

```
 Column |  Type   | Modifiers | Storage | Description
--------+---------+-----------+---------+-------------
 dn     | integer | not null  | plain   | ID
 mn     | integer | not null  | plain   | meaning
 ex     | integer | not null  | plain   | expression
Indexes:
    "dn_pkey" PRIMARY KEY, btree (dn)
    "dn_mn_key" UNIQUE, btree (mn, ex) CLUSTER
    "dn_ex_idx" btree (ex)
    "dn_mn_idx" btree (mn)
Foreign-key constraints:
    "dn_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
    "dn_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
Referenced by:
    TABLE "md" CONSTRAINT "md_dn_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)
    TABLE "pl0" CONSTRAINT "pl0_mn_fkey" FOREIGN KEY (mn, ex) REFERENCES dn(mn, ex) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "pl1" CONSTRAINT "pl1_mnex_fkey" FOREIGN KEY (mn, ex) REFERENCES dn(mn, ex) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "wc" CONSTRAINT "wc_dn_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)
Triggers:
    dnexap AFTER INSERT OR DELETE OR UPDATE ON dn FOR EACH ROW EXECUTE PROCEDURE exap()
```

9. Meaning Identifiers

A meaning may optionally have a (single) textual **meaning identifier**. It may serve to link the data in PanLex to a richer record in a source document.

Table “mi”

```
 Column |  Type   | Modifiers | Storage  | Description
--------+---------+-----------+----------+-------------
 mn     | integer | not null  | plain    | meaning
 tt     | text    | not null  | extended | text
Indexes:
    "mi_pkey" PRIMARY KEY, btree (mn) CLUSTER
Foreign-key constraints:
    "mi_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
```

Example: This meaning identifier allows access to taxonomic data about the meaning in the Arabic WordNet.

```
   mn    |          tt
---------+----------------------
 3613667 | Almmlkp_Almgrbyp_n1AR
```

<item itemid="Almmlkp_Almgrbyp_n1AR"
offset="108413097" lexfile="" name="المملكة المغربية"
type="synset" headword="" POS="n" source="NE file"
gloss="Morocco." authorshipid="1980" />
<authorship author="horacio" date="20070322" score=""
comment="NE import from file couma.out.
#Morocco#Kingdom of Morocco#" covering="0"
authorshipid="1980" />

10. Domain Descriptors

A meaning may optionally have **domain descriptors**. They are expressions. Attaching a domain descriptor to a meaning asserts that the meaning is within the domain described by the expression.

Table “dm”

```
 Column |  Type   | Modifiers | Storage | Description
--------+---------+-----------+---------+-------------
 dm     | integer | not null  | plain   | ID
 mn     | integer | not null  | plain   | meaning
 ex     | integer | not null  | plain   | expression
Indexes:
    "dm_pkey" PRIMARY KEY, btree (dm)
    "dm_mn_key" UNIQUE, btree (mn, ex) CLUSTER
Foreign-key constraints:
    "dm_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
    "dm_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
```

Example: This source says that “злучнік” in Belorusian and “союз” in Russian share a meaning in the domain (linguistics) described by “лінгв.” in Belorussian, which is expression 57303 in PanLex.)

```
  dm   |   mn    |  ex
-------+---------+-------
 38611 | 5149560 | 57303
```

злучальны	соединительный
злучнік	лінгв. союз
злучок	лінгв. дефис

11. Definitions

A meaning may optionally have **definitions**. A definition, which is a text in some variety, describes a meaning with more than a lexeme, so it is not a PanLex expression.

Table “df”

```
 Column |  Type   | Modifiers | Storage  |     Description
--------+---------+-----------+----------+---------------------
 df     | integer | not null  | plain    | ID
 mn     | integer | not null  | plain    | meaning
 lv     | integer | not null  | plain    | variety of the text
 tt     | text    | not null  | extended | text
Indexes:
    "df_pkey" PRIMARY KEY, btree (df)
    "df_mn_key" UNIQUE, btree (mn, lv, tt) CLUSTER
Foreign-key constraints:
    "df_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    "df_mn_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)
```

Example: A source document translates “couronne” in Cajun French into “crown” and “a wreath of flowers traditionally worn by a bride” in English. It is appropriate to treat “crown” as an expression and “a wreath … bride” as a definition in PanLex.

12. Word Classifications

A denotation may optionally have **word classifications**. These assign grammatical word classes (parts of speech) to denotations.

Table “wc”

```
 Column |  Type   | Modifiers | Storage |          Description
--------+---------+-----------+---------+-------------------------------
 wc     | integer | not null  | plain   | ID
 dn     | integer | not null  | plain   | denotation
 ex     | integer | not null  | plain   | PanLex word-class expression
Indexes:
    "wc_pkey" PRIMARY KEY, btree (wc)
    "wc_dn_key" UNIQUE, btree (dn, ex) CLUSTER
Foreign-key constraints:
    "wc_dn_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)
    "wc_ex_fkey" FOREIGN KEY (ex) REFERENCES wcex(ex) ON UPDATE CASCADE
```

12. Word Classifications

The word classes are drawn from a list.

Table “wcex”

```
                  Table "public.wcex"
 Column |  Type   | Modifiers | Storage  | Description
--------+---------+-----------+----------+-------------
 ex     | integer | not null  | plain    | expression
 tt     | text    | not null  | extended | text
Indexes:
    "wcex_pkey" PRIMARY KEY, btree (ex) CLUSTER
    "wcex_tt_key" UNIQUE, btree (tt)
Foreign-key constraints:
    "wcex_extt_fkey" FOREIGN KEY (ex, tt) REFERENCES pl0(ex, tt) DEFERRABLE INITIALLY DEFERRED
Referenced by:
    TABLE "wc" CONSTRAINT "wc_ex_fkey" FOREIGN KEY (ex) REFERENCES wcex(ex) ON UPDATE CASCADE
```

12. Word Classifications

The list of word classes is an extension of the list adopted by the Open Lexicon Interchange Format (OLIF) standard.

```
   ex     |  tt
----------+------
 3846607 | noun    noun
 3846608 | verb    verb
 3846609 | adjv    adjective
 3846610 | advb    adverb
 3846611 | name    proper noun
 3846614 | pron    pronoun
 3846615 | vpar    verb particle
 3956917 | auxv    auxiliary verb
 3956918 | detr    determiner
 3956920 | prep    preposition
 3956923 | post    postposition
 3956926 | conj    conjunction
 3956927 | ijec    interjection
 3956930 | affx    affix
 3957012 | misc    miscellaneous
```

OLIF

VALUE	DESCRIPTION
noun	noun
verb	verb
adj	adjective
adv	adverb
prep	preposition
conj	conjunction
det	determiner
part	verb particle
auxverb	auxiliary verb
pron	pronoun
punc	punctuation
other	other pos to be determined by user

13. Metadata

A denotation may optionally have **metadata**, consisting of variable-value pairs. Variables and values are arbitrary texts.

Table “md”

```
 Column |  Type   | Modifiers | Storage  | Description
--------+---------+-----------+----------+-------------
 md     | integer | not null  | plain    | ID
 dn     | integer | not null  | plain    | denotation
 vb     | text    | not null  | extended | variable
 vl     | text    | not null  | extended | value
Indexes:
    "md_pkey" PRIMARY KEY, btree (md)
    "md_dn_key" UNIQUE, btree (dn, vb, vl) CLUSTER
Foreign-key constraints:
    "md_dn_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)
```

Example: An English expression “pig” when synonymous with “police officer” could be annotated with a metadatum whose variable is “prag” and whose value is “vulg.”

14. Source Varieties

A source may optionally have **source varieties**. These are the language varieties that the source is described as documenting expressions in.

Table “av”

```
 Column |  Type   | Modifiers | Storage | Description
--------+---------+-----------+---------+-------------
 ap     | integer | not null  | plain   | source
 lv     | integer | not null  | plain   | variety
Indexes:
    "av_pkey" PRIMARY KEY, btree (ap, lv) CLUSTER
Foreign-key constraints:
    "av_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap) ON UPDATE CASCADE ON DELETE CASCADE
    "av_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv) ON UPDATE CASCADE ON DELETE CASCADE
```

15. Exemplar Characters

A language variety may optionally have **exemplar characters**. These are the characters that the Unicode Common Locale Data Repository designates as “exemplar characters” in the variety’s language. These are literal and quotation characters commonly encountered in expressions in the language.

Table “cu”

```
 Column |     Type      | Modifiers | Storage  |        Description
--------+---------------+-----------+----------+---------------------------
 lv     | integer       | not null  | plain    | variety
 c0     | character(5)  | not null  | extended | start of character range
 c1     | character(5)  | not null  | extended | end of character range
 loc    | text          |           | extended | locale
 vb     | text          | not null  | extended | variable
Indexes:
    "cu_c0_key" UNIQUE, btree (lv, c0, loc, vb) CLUSTER
    "cu_c1_key" UNIQUE, btree (lv, c1, loc, vb)
Foreign-key constraints:
    "cu_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
```

Comment: The start or end of a character range is represented with a 5-digit hexadecimal number. The “loc” field is a Unicode locale abbreviation, such as “Cyrl”. The “vb” field’s value is “pri” (primary) or “aux” (auxiliary).

16. Approved Characters

A language variety may optionally have **approved characters**.

Table “cp”

```
 Column |     Type     | Modifiers | Storage  |       Description
--------+--------------+-----------+----------+--------------------------
 lv     | integer      | not null  | plain    | variety
 c0     | character(5) | not null  | extended | start of character range
 c1     | character(5) | not null  | extended | end of character range
Indexes:
    "cp_pkey" PRIMARY KEY, btree (lv, c0) CLUSTER
    "cp_lv_key" UNIQUE, btree (lv, c1)
Foreign-key constraints:
    "cp_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
```

Comment: The start or end of a character range is represented with a 5-digit hexadecimal number.

Comment: Exemplar characters do not include digits or any punctuation except quotation marks. Approved characters are not subject to any such restriction.

17. Source Editors

A source may have a set of **permitted editors**.

Table “au”

```
 Column |  Type   | Modifiers | Storage |              Description
--------+---------+-----------+---------+---------------------------------------
 ap     | integer | not null  | plain   | source
 us     | integer | not null  | plain   | user permitted to edit the source
Indexes:
    "au_pkey" PRIMARY KEY, btree (ap, us)
Foreign-key constraints:
    "au_ap_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)
    "au_us_fkey" FOREIGN KEY (us) REFERENCES us(us)
```

18. Language Variety Editors

A language variety may have a set of **permitted editors**.

Table “lu”

```
 Column |  Type   | Modifiers | Storage |            Description
--------+---------+-----------+---------+-----------------------------------
 lv     | integer | not null  | plain   | language variety
 us     | integer | not null  | plain   | user permitted to edit the language variety
Indexes:
    "lu_pkey" PRIMARY KEY, btree (lv, us) CLUSTER
Foreign-key constraints:
    "lu_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
    "lu_us_fkey" FOREIGN KEY (us) REFERENCES us(us)
```

19. Design and Implementation Issues

A. Multiform expressions. Should PanLex recognize the phenomenon of expressions that have sets of forms related by transliteration, register, region, etc.? How would this affect the matching of added expressions with existing ones?

B. Lemmatic objects. Should PanLex include a table of unique lemmata (expression texts) and reference them in the table of expressions? Would this make the database more compact at the price of processing complexity?

C. New attributes. Should PanLex recognize additional attributes, such as pronunciations, inflections, etymologies, word subclasses (gender, aspect, declension, etc.), registers, argument frames, and usage examples? Would managing their complexity conflict with expanding the coverage of low-density languages?

D. Domain control. Should PanLex adopt a universal list of recognized domains? Would doing so prejudge a not yet consensual issue?

E. Attribute generalization. Should PanLex permit word classifications and metadata to be attached not only to denotations, but also to meanings and/or expressions? If to expressions, would this complicate the acquisition of additional data?

19. Design and Implementation Issues

F. Categorial metadata. Metadatum values are arbitrary, so may be elements of open sets such as pronunciations. For categorial values (e.g., “vulgar”), would a it be more useful to permit metadata whose values are PanLex expressions?

G. Valence. Should PanLex represent disapproval as well as approval? Should it be possible to represent the assertion that expressions A and B share no meaning? Or would quality ratings of sources substitute for this?

H. Confidence. Should PanLex represent probabilistic assertions? If so, what assertions should be eligible for this?

I. ID management. Should PanLex use serial generators as default ID values, instead of managing ID assignment? Would this simplification risk integer-range exhaustion in the table of denotations when large sources are repeatedly refreshed? Could this risk be easily eliminated with periodic daemonic recompaction?

J. Directedness. PanLex ignores translation directionality (if B is a translation of A, then A is a translation of B). Would the mandatory or optional assignment of a “head” property to one expression per meaning make PanLex more useful? If so, would the assignment of a “head” property to one language variety per source be granular enough?

Thanks

The database design of PanLex was inspired by the design of TransGraph, the sister (and original) project created by Kobi Reiter, Marcus Sammer, Michael Schmitz, Stephen Soderland, Oren Etzioni, and others at the Turing Center of the University of Washington.

Revisions from PanLex 1.7 to Panlex 2.0 were based in part on helpful suggestions made by members of the University of Washington Computational Linguistics Laboratory, at a presentation, “PanLex and TransGraph Schema Choices”, on 29 October 2008.

Numerous other valuable suggestions have been received from David Kamholz, Susan M. Colowick, Laura Welcher, Mausam, Oren Etzioni, PostgreSQL Experts, and others.